

[/dev/tbo](#)

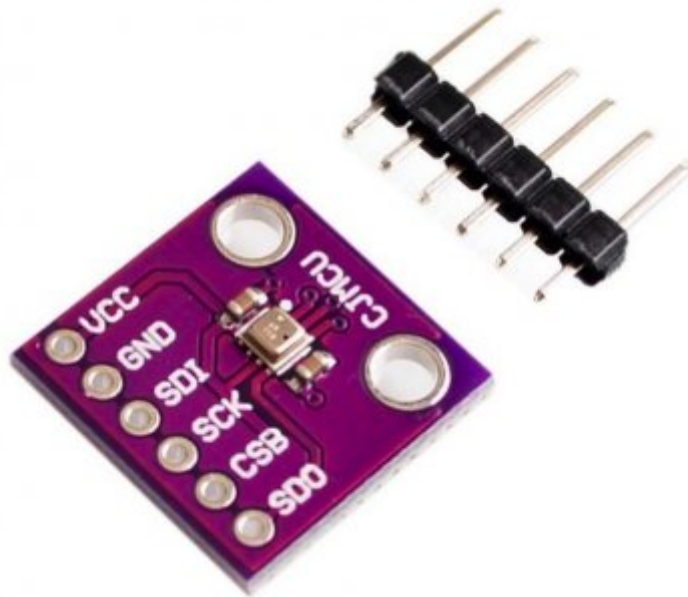
Langues du site [fr] français ▼

[Accueil](#) > [GNU/LINUX & RASPBERRY PI](#) > Raspberry & BME280 : mesure de température, pression et humidité (...)

# Raspberry & BME280 : mesure de température, pression et humidité relative

dimanche 6 novembre 2016, par [thebault](#)

## Caractéristiques du capteur



- documentation constructeur [https://ae-bst.resource.bosch.com/media/\\_tech/media/datasheets/BST-BMP280-DS001-12.pdf](https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP280-DS001-12.pdf)
- alimentation maxi : 3,6V
- mesure de la température de -40°C à 85°C avec une précision de +/- 1°C
- mesure de la pression de 300hPa à 1100 hPa avec une précision de +/- 1hPa (de 0°C à 65°C) ou +/- 1hPa (de -40°C à 0°C)
- mesure d'humidité relative de 0% à 100%
- sortie numérique sur bus I2C ou bus SPI

**Remarque** : dans la suite, nous allons utiliser le bus **I2C** pour communiquer avec le capteur.

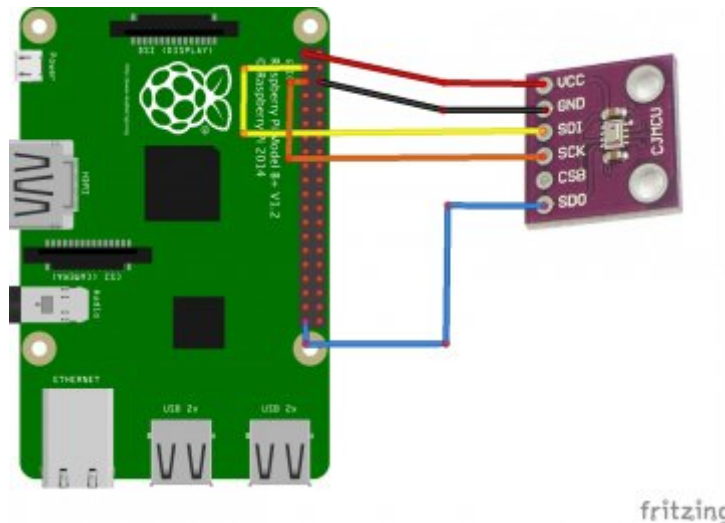
Rôle des broches.

Broche	fonction
Vcc	alimentation ( 3.6V max)
GND	masse
SDI	Données. A relier à SDA dans le cas du bus I2C
SCK	Horloge (clock). A relier à SCL dans le cas du bus I2C
CSB	permet de passer du mode SPI au mode I2C. Par défaut, CSB=1 donc mode I2C. Si CSB=0 au démarrage, alors le mode SPI est utilisé

Broche	fonction
SD0	choix de l'adresse I2C. Si SD0=0, alors l'adresse est 0x76, si SD0=1, alors l'adresse est 0x77

## Branchements

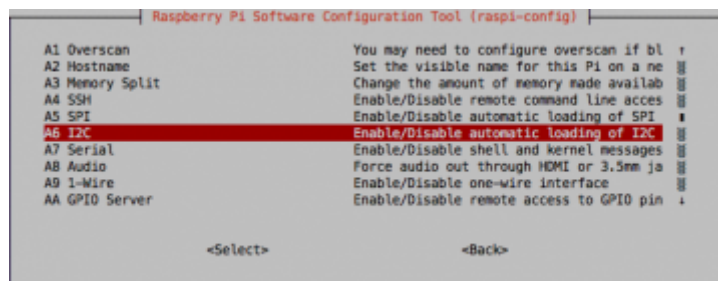
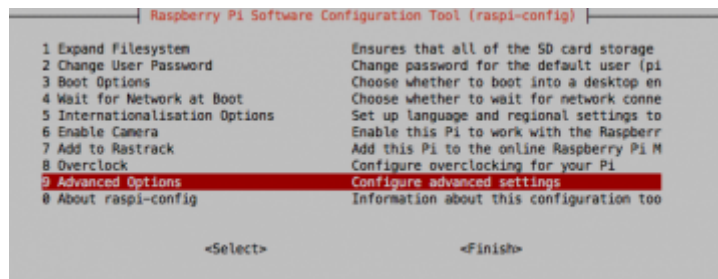
Dans les branchements suivants, SD0=0 donc l'adresse du capteur sur le bus I2C sera 0x76



## Installation sur debian Jessie

- activer le support i2c via raspi-config sur le raspberry Pi dans un terminal :  
`sudo raspi-config`

Aller dans "9 Advanced Options" puis "A6 I2C " puis "back" et "Finish" pour activer le support au niveau du noyau.



- installer les outils i2c pour la ligne de commande

I2ctools n'est pas indispensable mais il permet de vérifier la présence du capteur en ligne de commande

```
sudo apt-get install python-smbus
sudo apt-get install i2c-tools
```

- tester la présence du capteur

```
sudo i2cdetect -y 1
```

On obtient le résultat suivant, ce qui montre que l'adresse est bien 0x76 lorsque SD0=0 (relié à GND).

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- 76 -- -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

Si on avait relié SD0 à 3.3V (SD0=1), on aurait eu 0x77 pour adresse du capteur :

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- 77 -- -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

- installer le programme de test

On va utiliser le programme développée par Matt Hawkins <http://www.raspberrypi-spy.co.uk/2016/07/using-bme280-i2c-temperature-pressure-sensor-in-python/>

```
wget https://bitbucket.org/MattHawkinsUK/rpispys-misc/raw/master/python/bme280.py
```

Pour tester le programme (si le capteur est bien à l'adresse 0x76) avec 1 seule mesure.

```
python bme280.py
```

```
pi@raspberrypi:~/BME280 $ python bme280.py
Chip ID      : 96
Version      : 0
Temperature  : 20.83 C
Pressure     : 998.730443848 hPa
Humidity     : 58.7355708868 %
```

Si on souhaite utiliser l'adresse 0x77, il faut modifier la ligne suivante du programme bme.py : "**DEVICE** = 0x77 # Default device I2C address"

## Programme python

On va utiliser le programme précédent comme une bibliothèque.

Il faudra placer notre programme python dans le même répertoire que le fichier "bme.py" que l'on a téléchargé.

Pour cela, on ajoute "**import** bme280" dans notre programme.

Le programme suivant permet la mesure de la pression, température et humidité relative toute les 2 secondes.

```
1. # -*- coding: utf-8 -*-
2. import bme280
3. import time
4.
5. while True:
6.     temperature,pression,humidite = bme280.readBME280All()
```

```

7.      print "Temp : ", temperature, "°C \t P : ", pression, "hPa \t HR : ", humidite,
        "%"
8.      time.sleep(2)

```

[Télécharger](#)

```

pi@raspberrypi:~/BME280 $ python mesure1.py
Temp : 20.82 °C      P : 998.759280066 hPa      HR : 58.7352075611 %
Temp : 20.61 °C      P : 998.940587896 hPa      HR : 58.7251281566 %
Temp : 20.62 °C      P : 999.02495243 hPa      HR : 58.7256040683 %
Temp : 20.61 °C      P : 998.944659852 hPa      HR : 58.725249471 %
Temp : 20.62 °C      P : 998.952803786 hPa      HR : 58.7254920924 %
Temp : 20.62 °C      P : 998.939125873 hPa      HR : 58.7254920924 %
Temp : 20.63 °C      P : 998.996134792 hPa      HR : 58.7259679753 %
Temp : 20.63 °C      P : 999.00020691 hPa      HR : 58.7260892728 %

```

## Rubriques

- [ARDUINO](#)
- [DIVERS](#)
- [DIY Amps & fx](#)
- [GNU/LINUX & RASPBERRY PI](#)
- [MICROPYTHON](#)

Rechercher :  >>

## Dans la même rubrique

- [Utiliser la raspicam avec le raspberry PI](#)
- [Commande d'un servo moteur avec raspberry\\_pi et la bibliothèque WiringPi-python](#)
- [Raspberry & BME280 : mesure de température, pression et humidité relative](#)
- [Raspberry et carte PCA9685 16 sorties PWM](#)
- [Utilisation du GPIO avec Wiringpi-python](#)
- [un système GNU/Linux minimal sur RASPBERRY](#)
- [2ème partie. Commande PIFACE DIGITAL via interface web \[Raspberry PI + lighttpd + fastcgi + python\]](#)
- [1ère partie. Commande PIFACE DIGITAL via interface web \[Raspberry PI + lighttpd + CGI + python\]](#)
- [Raspi & Driver POLOLU DRV8835 Moteur à Courant Continu](#)

 | [Se connecter](#) | [Plan du site](#) |  [RSS 2.0](#)

Habillage visuel © [digitalnature](#) sous Licence [GPL](#)